25 | 异常设计，让错误无处遁形

2020-12-21 乔新亮

乔新亮的CTO成长复盘 进入课程 >

讲述：乔新亮
时长 14:57 大小 13.70M

你好，我是乔新亮。

今天，我想和你聊聊有关异常设计的话题。

如果你认真听了前面的内容，那么对你来说，异常设计应该不是一个新鲜概念了。在高可用设计、监控体系建设部分，我们都聊到了对异常的管理。

那么，为什么今天我们又要单独聊异常设计呢？因为异常管理虽然属于监控体系的一部分，但并不完全依赖于监控体系或高可用设计。**在企业的监控体系建设完成前，异常设一般就可以独立发挥作用，达到快速见效的目的。**

异常管理可以提升 IT 团队快速定位问题的能力、降低生产系统异常出现的频次和数量，更重要的是，它是让 IT 团队从面向技术，转为面向产品、面向用户的关键步骤，所以我们这里独立出来讲。

那么，在开始之前，我们有必要明确一下，究竟何为异常？

你可能会想，导致系统不能正常工作的问题就是异常呗。

说得对，但不全对。有些异常，可能当下未必会对生产环境产生影响，但不代表未来也不会；有些异常，可能在流量压力小时未必会出问题，但不代表流量压力大时也不会。

就像在平时，每位同学都会打印 Log，有 Warning，有 Error。大部分同学关注 Error，而不关注 Warning ，甚至有时 Error 也是选择性忽略的。反正打印这些，也不一定会导致线上服务受影响，看看就得了。

但很可能恰恰就是这些被忽视掉的 Error 和 Warning，最终导致了严重故障的出现。

所以，所谓**异常，指的就是那些让产品无法履行当初承诺用户的契约的问题。**

下面，我们通过几个具体的例子来理解一下。

那些被忽视掉的异常

早些年，经常网购的同学，可能听说过电商界的两个绰号：“二手 X”、“无货 X”。

商品缺货、出现品质或质量问题，一直以来就是电商界的两大“顽疾”。但严格地说，这些都是因仓储系统、订单系统、质量监测系统不够完善，而导致在业务运营层级发生的概率性问题，并非研发同学心目中的“恶性故障”。

但随着用户规模的增大，原本“不值一提”的低概率问题就会被无限放大。举个例子，一家电商平台缺货的概率可能只有 2%，也就是说，配货成功率高达 98%，看起来没啥问题。但当平台用户数量达到一百万时，就会有多达两万人受到缺货问题的困扰。这两万人中，可能只有 30% 的用户有在社交平台发帖的习惯。可即便如此，也会有 6000 人出现在各大社交平台上，发帖吐槽该电商平台的缺货、质量问题。

“好事不出门，坏事传千里”，于是，“无货 X”、“二手 X”的绰号就出现了，企业声誉严重受损，用户对企业的信任也极大降低，企业需要付出更大的努力来赢回用户。

所以，许多你不太关注的异常，最终会在产品层面对用户体验产生恶性影响。一般情况下，产品经理对这类问题的感知更敏锐，但职能型研发组织中的产品经理往往不参与代码编写，一般也不会深度参与开发过程；对于程序员来说，通常也不会主动和产品经理沟通类似问题。

于是，这类问题往往会顺利通过开发、测试，最终进入生产环境。在没有异常设计的情况下，让用户买单。**那些用户经历的坏体验，对于开发人员来说，可能只是一个简单的错误提示而已。**

当然，这是业务和产品层面的异常。对于很多普通技术人来说，对于底层技术层面的异常，可能更为熟悉。

我曾听一位做 C++ 服务器开发的同学，讲过他们企业发生的趣事。他们经常会处理许多到不同 CDN 节点的高频下载请求，但会有一部分下载超时或耗时增加的异常出现。

他们当时的做法是，当下载耗时超过 200ms 时，在 Log 中打印 Warning ；当下载因超时而失败时，设置定时重试，并在 Log 中打印 Error 。

我好奇地问，然后呢？

他回答，然后就没什么啦，重试一般都能成功。

我继续问，如果一直不成功呢？

他挠挠头，那可能就会导致业务出问题了，研发就得起床查 Log 了。

我又问，如果重试可以成功，但当并发压力增大时，来不及多次重试呢？

他尴尬地笑了下，没说话。

你看，往往我们在做所谓的“异常设计”时，并没有实现给用户的契约。对于电商平台而言，没有实现“成功下单 = 成功配送”的契约；对于那位做 C++ 服务器开发的同学来说，没有实现给其他系统模块的“准时下载”的契约。

大家通常认为，当下的异常没有导致生产环境出现严重问题，所以以后也不会出问题，但实际情况往往不是这样。

在我看来，没有兑现契约，即为异常，就应该控制起来。

那么，到底怎么做好异常设计呢？我们大致可以分为三部分来理解，分别是认知、方案和治理。

对于异常设计的认知、方案和治理

首先，对于异常设计，有五点认知一定要明确，分别是：

1. 异常一定要消灭：有异常，基本就意味着系统存在风险，一定要消灭异常；
2. 异常一定要管理：消灭异常是个长期工程，短期要通过管理行为来进行控制；
3. 对异常的处理水平，会极大影响产品的用户体验：用户规模越大，异常的影响往往越大；
4. 每个异常都要有具体的负责人：没有和具体的负责人一一对应，往往就意味着管理流于形式；
5. 与终端用户相关的异常，要以最高优先级处理：即便是 IT 研发，也要以用户为中心。

有了对异常的正确认知后，我们就需要在体系上对异常进行管理。

前面，在高可用设计部分，我曾经分享了“交易体系”和“协同体系”的设计差别。简单来说，交易体系负责处理相对稳定的业务流程，协同体系则是当异常出现时，需要接入的流程。比如客户要订 500 斤白菜，由交易体系来完成，库存满足、正常履行；库存不足时，则需要协同体系完成剩余流程。

但很多同学只做交易体系，不做协同体系，只抛出异常，而不处理异常，因此常常会引领用户走上“断头路”。在某些情况下，用户无路可走，只能找客服投诉。

认知到位后，下面我们一起来看如何落实异常设计。**异常设计一般包含异常注册、异常事件触发、异常协作流程以及异常统计。**

要管理好异常，我们首先要完成对异常的注册，注册内容大概分为以下几项：

1. 异常的 ID 和名字；
2. 对异常的描述；
3. 异常出现的代码位置；
4. 负责此异常的研发、测试和产品人员；
5. 异常发生时的代码版本；
6. 当时使用的异常处理程序。

同时，企业也要建设异常中心，各个系统都要在异常中心注册异常，一旦运行阶段出现异常，就要抛出异常事件，触发异常处理的协同流程。

当然，很多企业在还没有很好地管理异常时，就已经建设了大量的系统，因此很难再做异常设计。但很多异常都被记录在了日志里，因此研发同学可以通过收集日志中的异常，继而进行归类处理，再通过异常治理完成异常的注册，这样就间接达到了目标。

异常注册、收集后要做什么呢？当然是交由相关负责人进行处理了，这时我们就进入了异常的治理流程。不是所有的异常都要从 Log 中消失，但对于保留下的异常，一定提交管理层进行审批，说明保留原因；理由不够充分的，需要按排期规划并解决。

比如到 CDN 节点的下载行为有一定概率会失败，我们就要调查究竟是服务商提供的节点有问题，还是系统程序在编写方面有问题，又或者是相关机房的网络情况不好。总之，问题一定要明确并解决掉。

管理层也要关注异常的处理情况，包括：

1. 异常的数量；
2. 异常的发生频次；
3. 系统内异常数量的增速或降速；

……

在季度末、年末，我们会对管理层进行绩效考核，并将异常管理情况纳入考核体系，以此实现异常治理的闭环。

你可能会想，写个代码可真难啊，越来越麻烦了。不要怕麻烦，要关注投入产出比。做好异常设计，我们既可以做到高可用、高可靠问题的防微杜渐，还可以帮助研发和测试人员快速定位 Bug。最重要的是，在企业层面，我们可以**实现用户体验驱动内部经营完善的经营逻辑**。对于一个技术人来说，这无疑是能帮助你上台阶、再成长的。

在苏宁的时候，我主导研发了“神鉴”系统，目标就是为了做好企业的异常管理。在彩食鲜，我们也在不断完善异常管理，虽然时间不长，但相关工作已经卓有成效。前几天，相关人员还汇报道，我们科技中心所有产品的异常码都已经被统一管理起来，相关异常编码、响应信息都会被实时收集、归类，保证可视化、可统计。

你看，当我们能把繁杂的技术细节，纳入体系化的管理流程里时，还有什么样的困难我们克服不了呢？

当你用这种思路去管理异常时，就可以对一家企业有更深入、更近距离的了解。**只要仔细观察一家企业是如何对待异常的，就可以判断这家企业的精细化运营和管理水平。**

结语

今天，我们聊了聊与异常设计相关的话题，最终的目的，是让企业的潜在产品和研发问题，全部暴露出来，达成最好的产品体验。

但在落地异常设计时，也一定不要着急。要知道，异常，是会和我们长期共存的，对异常的处理和记录，也会逐渐成为企业数字资产的一部分。

对于直接影响用户体验的问题，要有契约精神，快速迭代；对于企业宏观角度的异常治理，要坚持长期主义，不断优化。

此外，企业的一个核心竞争力就是持续进化的能力。因此，在保持进化的过程中，企业发生的任何问题，都要纳入到异常管理流程中，将异常管理数据化、产品化、系统化，通过持续的治理和数据分析，让企业不断进化，最终建立竞争优势。

相信你一定能将异常管理做得越来越好，让一切错误无所遁形。

我们下一讲再见！

提建议

© 版权归极客邦科技所有，未经许可不得传播售卖。 页面已增加防盗追踪，如有侵权极客邦将依法追究其法律责任。

上一篇 24 | 监控设计，让一切都有迹可循，尽在掌控

精选留言 (9)

写留言

Weehua
2020-12-21

异常设计，这个概念还真是第一次听说，挺新鲜的。文中学习到了，如果能这样对待异常，系统化的管理异常，治理异常，我相信产品的质量和用户体验一定非常棒！

作者回复: 你好，Weehua
是的，企业如果一直坚持做异常治理，本身是持续完善、持续改进在产品层面的落实，一定会取得很好的效果，用户体验一定不会差。

1

狐狸糊涂
2020-12-21

感谢乔老师，每到周一三五就迫不及待的查看您更新的课程。有时可能在上班路上，有时可能是上班休息的时候。
乔老师，可以分享下您是怎么做知识储备的？或者可以分享您觉得不错的书籍吗？
展开

作者回复: 你好，狐狸糊涂
看书，交流，分享，实践。
早年多是看书，实践；现在交流的比重加大了些，越是到后来，越觉得认知的提升最重要，认知对了，其他其实都只是时间问题。

1 1

夜空中最亮的星（华仔...
2020-12-22

课程已经倒计时了啊
展开

rere
2020-12-22

不知不觉跟着乔老师来到了课程尾声，想趁着课程还没结束给乔老师留个言。

在试读乔老师第一课“五年上一个台阶”，就把那节课微信转给了十几个同事、甚至不同行业的朋友，也经常跟女朋友一起听、一起讨论。真的是发自内心的认可和分享，认为这一门课可以为在“业务导向型公司”的技术人员解困，找到真正可以努力的方向。...
展开

wst
2020-12-22

最重要的是，在企业层面，我们可以实现用户体验驱动内部经营完善的经营逻辑。这一点学习到了，让技术同学从技术视角转变为用户视角，进阶必备。

展开

spark

2020-12-21

异常设计是产品和研发微观能力的体现。乔老师分享说当访问规模和并发很大时，影响很大，把我吓到很难过。

产品经理如果把微观细节设计不好，开发兄弟们100%做不好异常设计开发，反正我以前的经历都是这样过来的。

团队中其他开发兄弟给我最大的压力就是这个异常设计问题，很多弟兄们技术比我好，...

展开

作者回复: 你好，spark

加油，你是非常认真学习的，太棒了

glutton

2020-12-21

乔老师好，认知又得到了提升

"异常，是导致不能履行契约的所有问题"

请教一个工程层面的问题，写代码时，如果当前已经捕获异常，有两种处理方式

1.在下层处理一下，包装成返回码；

2.直接抛出异常，由出口统一处理...

展开

作者回复: 你好，glutton

系统内，我选择2； 服务调用，我选择1

子夜枯灯

2020-12-21

一恍惚到了课程的结尾期了，感谢老乔一个多月的陪伴。如果您再开新的专栏我还会购买，记得要打折哦，(●ˇ∀ˇ●)。请教老乔一个问题，如何将自己碎片话的知识整理成体系？我再试图建立，但没有很好的技巧。期待您的回复。

展开

作者回复: 你好，子夜枯灯
可以去看看企业架构的书籍，根据那个体系进行整理

海
2020-12-21

乔总再三强调“契约精神”，我的理解就是做一个靠谱的工程师。

作者回复: 你好，海
做人，做企业都要有契约精神